Asymptotically Optimal Load Balancing in Large-scale Heterogeneous Systems with Multiple Dispatchers *

Xingyu Zhou
Department of ECE
The Ohio State University
Columbus, USA
zhou.2055@osu.edu

Ness Shroff
Department of ECE and CSE
The Ohio State University
Columbus, USA
shroff.11@osu.edu

Adam Wierman
Department of CMS
Caltech
Pasadena, USA
adamw@caltech.edu

ABSTRACT

We consider the load balancing problem in large-scale heterogeneous systems with multiple dispatchers. We introduce a general framework called Local-Estimation-Driven (LED). Under this framework, each dispatcher keeps local (possibly outdated) estimates of the queue lengths for all the servers, and the dispatching decision is made purely based on these local estimates. The local estimates are updated via infrequent communications between dispatchers and servers. We derive sufficient conditions for LED policies to achieve throughput optimality and delay optimality in heavy-traffic, respectively. These conditions directly imply delay optimality for many previous local-memory based policies in heavy traffic. Moreover, the results enable us to design new delay optimal policies for heterogeneous systems with multiple dispatchers. Finally, the heavy-traffic delay optimality of the LED framework also sheds light on a recent open question on how to design optimal load balancing schemes using delayed information.

Keywords

Asymptotically optimal; Load balancing; Heterogeneous systems, Multiple dispatchers, Delayed information

1. INTRODUCTION

Load balancing, which is responsible for dispatching jobs on parallel servers, has attracted significant interest in recent years. This is motivated by the challenges associated with efficiently dispatching jobs in large-scale data centers and cloud applications, which are rapidly increasing in size. A good load balancing policy not only ensures high throughput by maximizing server utilization, but improves the user experience by minimizing delay. There have been numerous load balancing policies proposed in the literature. The most straightforward one is Join-Shortest-Queue (JSQ), which has been shown to enjoy optimal delay in both non-asymptotic (for homogeneous servers) and asymptotic regimes [12, 1]. However, it is difficult to implement in today's large-scale data centers due to the large message overhead between the dispatcher and servers. As a result, alternative load balancing policies with low message overhead have been proposed. For example, the Power-of-d policy [6] has been shown to achieve optimal average delay in heavy traffic with only $2d$ messages per arrival [5]. Another common load balancing policy is the pull-based Join-Idle-Queue (JIQ) [4, 9], which has been shown to outperform the Power-of-d policy using less overhead. However, both Power-of-d and JIQ mainly achieve good performance for systems with homogeneous servers. Recently, some works consider heterogeneous servers and propose flexible and low message overhead policies that achieve optimal delay in heavy traffic [15, 14]. However, only a single dispatcher is considered in these works. Theoretical analysis of load balancing with multiple dispatchers has mainly focused on the JIQ policy so far [7, 10], which has a poor performance in heavy traffic and is even generally unstable for heterogeneous systems [15].

Note that heterogeneous systems with multiple dispatchers are now almost the default scenarios in today's cloud infrastructures. On one hand, the heterogeneity comes from the usage of multiple generations of CPUs and various types of devices [2]. On the other hand, with the massive amount of data, a scalable cloud infrastructure needs multiple dispatchers to increase both throughput and robustness [8].

Motivated by this, a recent work [11] proposes a new framework named Local Shortest Queue (LSQ) for designing load balancing policies for heterogeneous systems with multiple dispatchers. In particular, under this framework, each dispatcher keeps its own, local, and possibly outdated view of each server's queue length. Upon arrival, each dispatcher routes to the server with shortest local view. A small amount of message overhead is used to update the local view. The authors successfully establish sufficient conditions on the update scheme for the system to be stable. Moreover, extensive simulations were conducted to show that LSQ policies significantly outperform well-known low-communication policies while using similar communication overhead in both heterogeneous and homogeneous cases. However, no theoretical guarantee on the delay performance is provided and the authors mention it as an important future research direction. It is worth noting that the key challenge for establishing a delay performance guarantee for this framework is that it only uses possibly outdated local information to dispatch jobs. In fact, the problem of designing delay optimal load balancing schemes that only have access to delayed information has recently been listed as an open problem in [3].

Inspired by this, in this paper, we are particularly interested in the following questions: *Is is possible to establish delay performance guarantees for load balancing in hetero-*

*This project has been funded in part through NSF grants: CNS-2007231, CNS-1719371, and CNS-1717060 and NSF grants AitF-1637598 and CNS-1518941.

IFIP WG 7.3 Performance 2020 Milan, Italy

geneous systems with multiple dispatchers? If so, can these guarantees be achieved using only delayed information?

Contributions. To answer the questions above, we propose a general framework of load balancing for heterogeneous systems with multiple dispatchers that uses only delayed (out-of-date) information about the system state. We call this framework Local-Estimation-Driven (LED) and it generalizes the LSQ framework. Our main results provide sufficient conditions for LED policies to be both throughput optimal and delay optimal in heavy-traffic. Our key contributions can be summarized as follows.

First, we introduce the LED framework for designing load balancing policies for heterogeneous systems with multiple dispatchers. In this framework, each dispatcher keeps its own local estimates of queue lengths for all the servers, and makes its dispatching decision based purely on its own local estimates according to a certain dispatching strategy. The local estimates are updated infrequently via an update strategy that is based on communications between dispatchers and servers.

Second, we derive sufficient conditions for LED policies to be throughput optimal and delay optimal in heavy-traffic. The importance of the sufficient conditions is three-fold: (i) It can be shown that previous local-memory based policies (e.g., LSQ) satisfy our sufficient conditions. As a result, we are able to show that they are not only throughput optimal (in a stronger sense) but also delay optimal in heavy-traffic. (ii) The conditions allow us to design new delay optimal load balancing policies with zero dispatching delay and low message overhead that work for heterogeneous servers and multiple dispatchers. (iii) These conditions also provide us with a systematic approach for generalizing previous optimal policies to the case of multiple dispatchers and exploring the trade-off between memory (i.e., local estimations) and message overhead. For instance, we are able to show that the Power-of-d policy can achieve delay optimality in heavy traffic, even in heterogeneous systems, as long as the imbalance among the service rates is not too large.

Third, the LED framework also sheds light on the open problem posed in [3], which asks how to design heavy-traffic delay optimal policies that only use delayed information. Our main results for LED policies not only demonstrate that it is possible to achieve optimal delay in heavy-traffic via only delayed information, but highlight conditions on the extent to which old information is useful. Moreover, they provide methods for using the delayed information to achieve optimality in heavy traffic. Interestingly, the LED framework also shows that, in the case of multiple dispatchers, inaccurate information can actually lead to improved performance.

To establish the main results, we need to address the following two technical challenges. First, each dispatcher in our model only has access to delayed and outdated system information. Second, we consider a large class of dispatching strategies specified by a general condition. To handle the general condition, we have to apply a refined drift analysis to obtain the necessary negative drifts required for throughput optimality and delay optimality. In order to handle the outdated queue length information, we have to transfer the drift on local estimates to the corresponding drift on the actual queue lengths. To this end, we develop a new Lyapunov function, and combine this with sample-path analysis, and couplings arguments to obtain tight bounds.

The full paper is available at [13].

2. REFERENCES

[1] Atilla Eryilmaz and R Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, 72(3-4):311–359, 2012.

[2] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. Evolve or die: High-availability design principles drawn from googles network infrastructure. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 58–72, 2016.

[3] David Lipshutz. Open problemload balancing using delayed information. *Stochastic Systems*, 9(3):305–306, 2019.

[4] Yi Lu, Qiaomin Xie, Gabriel Kliot, Alan Geller, James R Larus, and Albert Greenberg. Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation*, 68(11):1056–1071, 2011.

[5] Siva Theja Maguluri, R Srikant, and Lei Ying. Heavy traffic optimal resource allocation algorithms for cloud computing clusters. *Performance Evaluation*, 81:20–39, 2014.

[6] Michael Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.

[7] Michael Mitzenmacher. Analyzing distributed join-idle-queue: A fluid limit approach. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pages 312–318. IEEE, 2016.

[8] Patrick Shuff. Building a billion user load balancer. 2016.

[9] Alexander L Stolyar. Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems*, 80(4):341–361, 2015.

[10] Alexander L Stolyar. Pull-based load distribution among heterogeneous parallel servers: the case of multiple routers. *Queueing Systems*, 85(1-2):31–65, 2017.

[11] Shay Vargaftik, Isaac Keslassy, and Ariel Orda. Lsq: Load balancing in large-scale heterogeneous systems with multiple dispatchers. *IEEE/ACM Transactions on Networking*, 2020.

[12] Richard R Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, pages 406–413, 1978.

[13] Xingyu Zhou, Ness Shroff, and Adam Wierman. Asymptotically optimal load balancing in large-scale heterogeneous systems with multiple dispatchers. *arXiv preprint arXiv:2002.08908*, 2020.

[14] Xingyu Zhou, Jian Tan, and Ness Shroff. Heavy-traffic delay optimality in pull-based load balancing systems: Necessary and sufficient conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(3):1–33, 2018.

[15] Xingyu Zhou, Fei Wu, Jian Tan, Yin Sun, and Ness Shroff. Designing low-complexity heavy-traffic delay-optimal load balancing schemes: Theory to algorithms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):39, 2017.